Refine Search

Search Results -

Terms	Documents
L11 AND (microcode or microprogram)	0

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L13

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Thursday, March 17, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=NO; OP=OR			
L13	L11 AND (microcode or microprogram)	0	L13
L12	L11 and (validation or verification) near (software or program)	0	L12
L11	(antilock ADJ brakes) Or (antilock ADJ brake ADJ system) AND software	101	L11
L10	L5 AND microcode	48	L10
L9	L5 and software.ab.	17	L9
L8	L5 AND (program near verification)	0	L8
L7	L5 AND security	140	L7
L6	L5 AND (serial or sequential)	282	L6
L5	L4 AND verification	428	L5
L4	L3 AND software	4175	L4
L3	ABS OR (antilock ADJ brakes) Or (antilock ADJ brake ADJ system) AND software	53673	L3

L2	L1 ANd 713/$$$.ccls.	8	L2
L1	FIBs	344	L1

END OF SEARCH HISTORY

Refine Search

Search Results -

Terms	Documents
5347581.pn. or 5452357.pn. or 6698662.pn.	3

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L1

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Thursday, March 17, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=NO; OP=OR		
L1	5347581.pn. or 5452357.pn. or 6698662.pn.	3	L1

END OF SEARCH HISTORY

Double patenting search

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 3 of 3 returned.

☑ 1. Document ID: US 6698662 B1

L1: Entry 1 of 3 File: USPT Mar 2, 2004

US-PAT-NO: 6698662
DOCUMENT-IDENTIFIER: US 6698662 B1

TITLE: Devices for hiding operations performed in a microprocesser card

DATE-ISSUED: March 2, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Feyt; Nathalie	Marseilles			FR
Benoit; Olivier	Aubagne			FR
Naccache; David	Paris			FR

US-CL-CURRENT: 235/492; 235/382.5

ABSTRACT:

The invention concerns microprocessor cards and, in such cards, various devices for hiding operations performed in the card against fraudulent breaches by analysing the current consumed. The invention is characterised in that it consists in adding in the card a device (20) modifying the consumed current, either by averaging it by integration, or by adding thereto random values by a random signal generator (28) so as to hide the operations performed. In another embodiment, it consists in carrying out simultaneously an operation for making secure and writing in an EEPROM memory, the latter generating chaotic current variations which hide the operation to be made secure.

6 Claims, 3 Drawing figures
Exemplary Claim Number: 2
Number of Drawing Sheets: 3

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☑ 2. Document ID: US 5452357 A

L1: Entry 2 of 3 File: USPT Sep 19, 1995

US-PAT-NO: 5452357
DOCUMENT-IDENTIFIER: US 5452357 A

TITLE: Method and apparatus for access control and/or identification

DATE-ISSUED: September 19, 1995

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Naccache; David	Maisons-Alfort			FR

US-CL-CURRENT: 713/172; 380/215, 380/229, 380/30

ABSTRACT:

A system providing access control, including encryption and decryption capability, replaces a public key directory by a transmission between an authority, or a sender S, and a receiver R of a "seed" value. The seed is processed to provide both identity information for R and public keys, i.e. a "virtual public key directory", or VPKD. The VPKD is generated prior to execution of the algorithm requiring the public directory, i.e. the host algorithm.

28 Claims, 8 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

☑ 3. Document ID: US 5347581 A

L1: Entry 3 of 3 File: USPT Sep 13, 1994

US-PAT-NO: 5347581
DOCUMENT-IDENTIFIER: US 5347581 A

TITLE: Verification process for a communication system

DATE-ISSUED: September 13, 1994

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Naccache; David	Maisons-Alfort			FR
M'Raihi; David	Paris			FR

US-CL-CURRENT: 380/30; 380/28, 713/155, 713/168, 713/177, 713/180

ABSTRACT:

The process concerns a system, comprising communication devices A1, A2, . . . A.alpha. connected to a central verification device B by the means of communication interfaces wherein each device Ai, having data processing means, communication means, memory means and random or pseudo-random generation means, transmits to the device B, having data processing means, communication means and memory means, a set of DSS digital signatures. Once all the signatures has been received by device B, device B verifies them simultaneously by performing few calculations for verifying

a great same number of signatures sequentially.

7 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
5347581.pn. or 5452357.pn. or 6698662.pn.	3

Display Format: REV Change Format

Previous Page Next Page Go to Doc#

Day : Thursday
Date: 3/17/2005

PALM INTRANET

Time: 09:21:31

Inventor Name Search Result

Your Search was:

Last Name = ROUSSEAU
First Name = LUDOVIC

Application#	Patent#	Status	Date Filed	Title	Inventor Name
09763159	Not Issued	161	08/16/1999	METHOD AND DEVICE FOR AUTHENTICATING WITH SYMMETRICAL ALGORITHM	ROUSSEAU, LUDOVIC
09936174	Not Issued	030	12/06/2001	METHOD FOR MONITORING A PROGRAM FLOW	ROUSSEAU, LUDOVIC
10311698	Not Issued	030	05/30/2003	ACCESS CONTROL TO DATA PROCESSING MEANS	ROUSSEAU, LUDOVIC

Inventor Search Completed: No Records to Display.

Search Another: Inventor **Last Name** ROUSSEAU **First Name** LUDOVIC Search

To go back use Back button on your browser toolbar.

Back to PALM | ASSIGNMENT | OASIS | Home page

Day : Thursday
Date: 3/17/2005

PALM INTRANET

Time: 09:16:08

Inventor Name Search Result

Your Search was:

Last Name = NACCACHE
First Name = DAVID

Application#	Patent#	Status	Date Filed	Title	Inventor Name
08082228	5414772	150	06/23/1993	SYSTEM FOR IMPROVING THE DIGITAL SIGNATURE ALGORITHM	NACCACHE, DAVID
08094058	5502764	150	01/24/1994	METHOD, IDENTIFICATION DEVICE AND VERIFICATION DEVICE FOR IDENTIFICATION AND/OR PERFORMING DIGITAL SIGNATURE	NACCACHE, DAVID
08094931	Not Issued	166	07/16/1993	METHOD AND APPARATUS FOR ACCESS CONTROL AND/OR IDENTIFICATION	NACCACHE, DAVID
08121599	5461675	150	09/14/1993	APPARATUS AND METHOD FOR ACCESS CONTROL	NACCACHE, DAVID
08122716	5347581	150	09/15/1993	VERIFICATION PROCESS FOR A COMMUNICATION SYSTEM	NACCACHE, DAVID
08135491	5654891	150	10/13/1993	METHOD AND APPARATUS FOR CONTROLLING AND/OR LIMITING SPEED EXCESS BY DRIVERS	NACCACHE, DAVID
08135492	Not Issued	163	10/13/1993	CARD, CARD READER AND METHOD FOR PROTOCOL SELECTION	NACCACHE, DAVID
08135493	5434917	150	10/13/1993	UNFORGEABLE IDENTIFICATION DEVICE, IDENTIFICATION DEVICE READER AND METHOD OF IDENTIFICATION	NACCACHE, DAVID
08199213	5479511	150	07/08/1994	METHOD, SENDER APPARATUS AND RECEIVER APPARATUS FOR MODULO OPERATION	NACCACHE, DAVID

08211280	Not Issued	167	11/16/1994	METHOD AND APPARATUS FOR SECURE TRANSMISSION OF VIDEO SIGNALS	NACCACHE, DAVID
08211281	Not Issued	161	11/07/1994	METHOD AND APPARATUS FOR SECURE TRANSMISSION OF VIDEO SIGNALS	NACCACHE, DAVID
08347570	5625695	150	11/30/1994	PROCESS FOR GENERATING DSA SIGNATURES WITH LOW-COST PORTABLE APPARATUSES	NACCACHE, DAVID
08400310	5452357	150	03/06/1995	METHOD AND APPARATUS FOR ACCESS CONTROL AND/OR IDENTIFICATION	NACCACHE, DAVID
08765483	5910989	150	03/06/1997	METHOD FOR THE GENERATION OF ELECTRONIC SIGNATURES, IN PARTICULAR FOR SMART CARDS	NACCACHE, DAVID
08860832	5946397	150	09/24/1997	METHOD OF CRYPTOGRAPHY WITH PUBLIC KEY BASED ON THE DISCRETE LOGARITHM	NACCACHE, DAVID
08875331	6226382	150	09/24/1997	METHOD FOR IMPLEMENTING A PRIVATE-KEY COMMUNICATION PROTOCOL BETWEEN TWO PROCESSING DEVICES	NACCACHE, DAVID
09194980	6459791	150	08/24/1999	PUBLIC KEY CRYPTOGRAPHY METHOD	NACCACHE, DAVID
09308369	Not Issued	161	07/13/2000	METHOD FOR SIGNING AND/OR AUTHENTICATION ELECTRONIC MESSAGES	NACCACHE, DAVID
09377666	Not Issued	161	08/19/1999	A CRYPTOGRAPHIC SYSTEM COMPRISING AN ENCRYPTION AND DECRYPTION SYSTEM AND A KEY ESCROW STSTEM AND THE ASSOCIATED EQUIPMENT AND DEVICES	NACCACHE, DAVID
09434102	Not Issued	161	11/05/1999	PSEUDO-RANDOM GENERATOR BASED ON A HASH CODING FUNCTION FOR CRYPTOGRAPHIC SYSTEMS REQUIRING RANDOM DRAWING	NACCACHE, DAVID
09646564	6698662	150	12/18/2000	DEVICES FOR HIDING	NACCACHE,

				OPERATIONS PERFORMED IN A MICROPROCESSER CARD	DAVID
09763158	Not Issued	071	05/07/2001	METHOD FOR TESTING A RANDOM NUMBER SOURCE AND ELECTRONIC DEVICES COMPRISING SAID METHOD	NACCACHE, DAVID
09802968	Not Issued	041	03/12/2001	PROBABILISTIC DIGITAL SIGNATURE METHOD	NACCACHE, DAVID
09807614	Not Issued	030	07/11/2001	ELECTRONIC COMPONENT FOR MASKING EXECUTION OF INSTRUCTIONS OR DATA MANIPULATION	NACCACHE, DAVID
09936174	Not Issued	030	12/06/2001	METHOD FOR MONITORING A PROGRAM FLOW	NACCACHE, DAVID
09959944	Not Issued	030	02/20/2002	COUNTERMEASURE METHOD IN AN ELECTRONIC COMPONENT USING A DYNAMIC SECRET KEY CRYPTOGRAPHIC ALGORITHM	NACCACHE, DAVID
10031065	Not Issued	030	04/02/2002	METHOD FOR IMPROVING A RANDOM NUMBER GENERATOR TO MAKE IT MORE RESISTANT AGAINST ATTACKS BY CURRENT MEASURING	NACCACHE, DAVID
10048216	Not Issued	030	04/25/2002	SIGNATURE SCHEMES BASED ON DISCRETE LOGARITHM WITH PARTIAL OR TOTAL MESSAGE RECOVERY	NACCACHE, DAVID
10130937	Not Issued	030	05/24/2002	A METHOD FOR ENCODING LONG MESSAGES FOR ELECTRONIC SIGNATURE SCHEMES BASED ON RSA	NACCACHE, DAVID
10130943	Not Issued	030	05/24/2002	PROTECTION AGAINST THE IMPROPER USE OF AN INSTRUCTION IN A MEMORY	NACCACHE, DAVID
10148022	Not Issued	030	05/24/2002	METHOD FOR ACCELERATED TRANSMISSION OF ELECTRONIC SIGNATURE	NACCACHE, DAVID
10257130	Not Issued	030	10/09/2002	METHOD FOR CALCULATING CRYPTOGRAPHIC KEY CHECK DATA	NACCACHE, DAVID

Poss DP POV

10257411	Not Issued	020	10/11/2002	COUNTERMEASURE METHOD IN A MICROCIRCUIT, MICCROCIRCUIT THEREFORE AND SMART CARD COMPRISING SAID MICROCIRCUIT	NACCACHE, DAVID
10311698	Not Issued	030	05/30/2003	ACCESS CONTROL TO DATA PROCESSING MEANS	NACCACHE, DAVID
10398732	Not Issued	030	04/09/2003	METHOD FOR PROTECTION AGAINST FRAUD IN A NETWORK BY ICON SELECTION	NACCACHE, DAVID
10467718	Not Issued	030	08/12/2003	METHOD FOR MULTIPLYING TWO BINARY NUMBERS	NACCACHE, DAVID
10467928	Not Issued	020	03/22/2004	IDENTIFICATION MODULE PROVIDED WITH A SECURE AUTHENTICATION CODE	NACCACHE, DAVID
10817453	Not Issued	020	04/05/2004	CRYPTOGRAPHIC SYSTEM COMPRISING AN ENCRYPTION AND DECRYPTION SYSTEM AND A KEY ESCROW SYSTEM, AND THE ASSOCIATED EQUIPMENT AND DEVICES	NACCACHE, DAVID
10859978	Not Issued	030	06/04/2004	BIOMETRIC IDENTIFICATION METHOD AND DEVICE ADAPTED TO VERIFICATION ON CHIP CARDS	NACCACHE, DAVID

Inventor Search Completed: No Records to Display.

Search Another: Inventor **Last Name** NACCACHE **First Name** DAVID Search

To go back use Back button on your browser toolbar.

Back to PALM | ASSIGNMENT | OASIS | Home page

Day : Thursday
Date: 3/17/2005

PALM INTRANET

Time: 09:21:42

Inventor Name Search Result

Your Search was:

Last Name = GIRARD
First Name = PIERRE

Application#	Patent#	Status	Date Filed	Title	Inventor Name
06088670	4325055	150	10/26/1979	ANALOG-TO-DIGITAL CONVERTER	GIRARD, PIERRE
06129656	4331713	150	03/12/1980	PROCESS AND APPARATUS FOR THE CONTINUOUS COATING OF A SHEET ARTICLE, PARTICULARLY A WEB OF PAPER OR PAPERBOARD	GIRARD, PIERRE
06323312	Not Issued	161	11/20/1981	PROCESS AND APPARATUS FOR THE CONTINUOUS COATING OF A SHEET ARTICLE PARTICULARLY A WEB OF PAPER OR PAPERBOARD	GIRARD, PIERRE
06680991	Not Issued	163	12/13/1984	PROCESS AND APPARATUS FOR THE CONTINUOUS COATING OF A SHEET ARTICLE, PARTICULARLY A WEB OF PAPER OR PAPERBOARD	GIRARD, PIERRE
07591378	5167226	150	10/01/1990	COMBINED CLAPPING AND VIBRATING DEVICE FOR EXPELLING RETAINED OBSTRUCTIVE SECRETIONS IN THE LUNGS	GIRARD, PIERRE
08683851	5827616	150	07/19/1996	COATED GREASEPROOF PAPER AND PROCESS FOR MANUFACTURING IT	GIRARD, PIERRE
08886144	6144406	150	06/30/1997	ELECTRONIC PANORAMIC CAMERA	GIRARD, PIERRE
08988064	5965091	150	12/10/1997	FILLED PAPER FOR GAS FILTRATION	GIRARD, PIERRE
08988066	6224768	150	12/10/1997	FILTER PAPER FOR LADEN LIQUIDS	GIRARD, PIERRE

08988067	5984110	150	12/10/1997	DEVICE FOR PURIFYING LIGHTLY LADEN WATER	GIRARD, PIERRE
09508316	Not Issued	041	05/26/2000	PAPER OR CARDBOARD WITH IMPROVED PRINTABILITY	GIRARD, PIERRE
09647650	Not Issued	094	10/03/2000	PHOTOCATALYTIC COMPOSITION	GIRARD, PIERRE
09743187	6503447	150	01/05/2001	METHOD FOR PURIFYING GASEOUS EFFLUENTS BY MEANS OF PHOTOCATALYSIS, INSTALLATION FOR CARRYING OUT SAID METHOD	GIRARD, PIERRE
09936174	Not Issued	030	12/06/2001	METHOD FOR MONITORING A PROGRAM FLOW	GIRARD, PIERRE
10181053	Not Issued	071	10/10/2002	METHOD FOR PROTECTING AGAINST THEFT OF A PIN NUMBER IN (A) MULTI-APPLICATION SMART CARD (S) AND CHIP CARD(S) IMPLEMENTING SAID METHOD	GIRARD, PIERRE
10181884	Not Issued	030	10/10/2002	METHOD FOR PROTECTING AGAINST THEFT THE AUTHENTICATING VALUE OF MULTIPLE APPLICATION SMART CARDS, SMART CARDS THEREFOR AND TERMINALS DESIGNED TO RECEIVE SAID CARDS	GIRARD, PIERRE
10276920	Not Issued	030	01/03/2003	METHOD FOR PROTECTION AGAINST FRAUDULENT MODIFICATION OF DATA SENT TO A SECURE ELECTRONIC MEDIUM	GIRARD, PIERRE
10296547	Not Issued	030	11/25/2002	MAKING SECURE DATA EXCHANGES BETWEEN CONTROLLERS	GIRARD, PIERRE
10311698	Not Issued	030	05/30/2003	ACCESS CONTROL TO DATA PROCESSING MEANS	GIRARD, PIERRE
10343112	Not Issued	030	01/28/2003	METHOD FOR MAKING SECURE A SESSION WITH DATA PROCESSING MEANS UNDER THE CONTROL OF SEVERAL ENTITIES	GIRARD, PIERRE

10467763	Not Issued	019	01/01/0001	DYNAMIC MANAGEMENT OF ACCESS RIGHTS LISTS IN A PORTABLE ELECTRONIC OBJECT	GIRARD, PIERRE
10484524	Not Issued	020	09/20/2004	METHOD FOR PROTECTING PERSONAL DATA READ IN A TERMINAL STATION BY A SERVER	GIRARD, PIERRE
10520434	Not Issued	019	01/01/0001	MAKING SECURE DOWNLOAD APPLICATION IN PARTICULAR IN A SMART CARD	GIRARD, PIERRE
10838023	Not Issued	095	05/03/2004	PHOTOCATALYTIC COMPOSITION	GIRARD, PIERRE
06141425	4353504	150	04/18/1980	HIGH PRESSURE SNOW GUN	GIRARDIN, PIERRE
06802043	4717072	150	11/26/1985	SEQUENTIAL VALVE DRAIN FOR SNOW GUN	GIRARDIN, PIERRE
07246667	4914923	150	08/05/1988	METHOD OF COVERING ARTIFICIAL ALPINE- OR NORDIC-SKIING TRACKS WITH SNOW AND MEANS FOR IMPLEMENTING THE METHOD	GIRARDIN, PIERRE
08501047	Not Issued	169	08/19/1995	IMPROVED SNOW GUN	GIRARDIN, PIERRE
08513850	Not Issued	169	09/12/1995	SUPPORT FOR A PULVERISATION APPARATUS FOR A MIXTURE OF WATER AND AIR UNDER PRESSURE	GIRARDIN, PIERRE
08525658	Not Issued	161	09/29/1995	SPRAYING NOZZLE AND DEVICE FOR SPRAYING A MIXTURE OF WATER AND USING SAID NOZZLE	GIRARDIN, PIERRE

Inventor Search Completed: No Records to Display.

Search Another: Inventor **Last Name** GIRARD **First Name** PIERRE Search

To go back use Back button on your browser toolbar.

Back to PALM | ASSIGNMENT | OASIS | Home page

Refine Search

Search Results -

Terms	Documents
L1 and (commands or opcodes)	10

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L2

Refine Search | Recall Text | Clear | Interrupt

Search History

DATE: Thursday, March 17, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=NO; OP=OR		
L2	L1 and (commands or opcodes)	10	L2
L1	717/127.ccls. and (microcode or assembler)	22	L1

END OF SEARCH HISTORY

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 10 of 10 returned.

☐ 1. Document ID: US 6754888 B1

L2: Entry 1 of 10 File: USPT Jun 22, 2004

US-PAT-NO: 6754888
DOCUMENT-IDENTIFIER: US 6754888 B1
** See image for Certificate of Correction **

TITLE: Facility for evaluating a program for debugging upon detection of a debug trigger point

DATE-ISSUED: June 22, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dryfoos; Robert O.	Poughkeepsie	NY		
Matela, Jr.; Richard J.	Pawling	NY		
Sutton; Leslie F.	Lagrangeville	NY		

US-CL-CURRENT: 717/127; 717/128, 717/130

ABSTRACT:

Method, system and program products for screening a program of a computing environment for debug processing are presented which employ a table that contains information identifying at least some program areas of storage of the computing environment where programs to be debugged may reside and information identifying programs of the computing environment to be excluded from debugging. After detecting a debug trigger point during execution of a program, screening is accomplished by referencing the table to first ascertain whether the trigger point is within a program area of storage meaning that the program under execution may comprise an application program to be debugged, and if so, referencing the table to determine whether the program has been explicitly excluded from the debug processing. If not excluded, debug processing may proceed.

16 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 3

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☐ 2. Document ID: US 6718286 B2

L2: Entry 2 of 10 File: USPT Apr 6, 2004

US-PAT-NO: 6718286
DOCUMENT-IDENTIFIER: US 6718286 B2

TITLE: Non-intrusive application code profiling method and apparatus

DATE-ISSUED: April 6, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Rivin; Russell L.	Holliston	MA		
Bellavance; Lori A.	North Attleboro	MA		

US-CL-CURRENT: 702/186; 717/127, 717/128, 717/131

ABSTRACT:

System and method for monitoring a processor when it executes software code for a computer program. A register collects information regarding instructions executed by the processor, from the program counter; and a sampler, operatively connected to the register, asynchronously from the operation of the processor, samples contents of the register. The sampler may provide the samples to a host computer via a shift register in a JTAG port, and the host computer may provide a statistical record of the instructions executed by the processor.

32 Claims, 2 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 2

☐ 3. Document ID: US 6658651 B2

L2: Entry 3 of 10 File: USPT Dec 2, 2003

US-PAT-NO: 6658651
DOCUMENT-IDENTIFIER: US 6658651 B2

TITLE: Method and apparatus for analyzing software in a language-independent manner

DATE-ISSUED: December 2, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
O'Brien; Stephen Caine	Tigard	OR		
Maxwell, III; Sidney R.	Bothell	WA		

US-CL-CURRENT: 717/127

ABSTRACT:

A software analysis system for capturing tags generated by tag statements in instrumented source code. The system includes a probe that monitors the address and data bus of the target system. When a tag statement is executed in the target system, a tag is written to a predetermined location in the address space of the target. The tag contains a tag value that is indicative of the location in the source code of the tag statement generating the tag. The source code instrumenter includes a language-dependent parser and a language-independent analyzer that records tagging data in a symbol database. The system performs a variety of analysis functions in essentially real time, including code coverage, function and task execution times, memory allocation, call pairs, and program tracing.

9 Claims, 19 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 19

☐ 4. Document ID: US 6467082 B1

L2: Entry 4 of 10 File: USPT Oct 15, 2002

US-PAT-NO: 6467082
DOCUMENT-IDENTIFIER: US 6467082 B1

TITLE: Methods and apparatus for simulating external linkage points and control transfers in source translation systems

DATE-ISSUED: October 15, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
D'Arcy; Paul Gerard	Alpharetta	GA		
Deschler; Pamela C.	Bethlehem	PA		
Jinturkar; Sanjay	Bethlehem	PA		
Peri; Kamesh	Allentown	PA		
Peri; Ramesh V.	Allentown	PA		
Whalley; David B.	Tallahassee	FL		

US-CL-CURRENT: 717/127; 717/136, 717/163

ABSTRACT:

A method for simulating a first processor (e.g., target processor) on a second processor (e.g., host processor) includes translating assembly language instructions associated with the first processor into `C` language code. The `C` language code is then compiled by a compiler program running on the second processor. The compiled code is then executed by the second processor to simulate the first processor. For example, the code may be checked to determine whether it is functionally correct and/or run-time statistics may be collected regarding the program associated with the first processor.

40 Claims, 15 Drawing figures

Exemplary Claim Number: 1
Number of Drawing Sheets: 12

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☐ 5. Document ID: US 6393606 B1

L2: Entry 5 of 10 File: USPT May 21, 2002

US-PAT-NO: 6393606
DOCUMENT-IDENTIFIER: US 6393606 B1

TITLE: Inverse assembler

DATE-ISSUED: May 21, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Davila; Marco A.	Colorado Springs	CO		
Bunker; Christopher W.	Colorado Springs	CO		
Bernard; Christopher T.	Colorado Springs	CO		

US-CL-CURRENT: 717/127; 703/23, 717/131, 717/134, 717/136, 717/141

ABSTRACT:

An inverse assembler and converter acquire binary code during inverse assembly of compiled programming code for a software application. A memory image file is generated during compiling of the programming code and a converter is used to trigger a physical address in a memory bus via a logic analyzer. A triggered logical address in the compiled programming code is determined. The triggered logical address is input into the converter. Trigger commands are provided to the logic analyzer that are used to trigger a physical address where binary code is stored in memory. The trigger commands are supplied to the logic analyzer and the memory bus is triggered. The physical address is acquired and converted into a logical address. The memory image file is searched for the logical address. The binary code is acquired from the memory image file at the logical address. The binary code corresponds to only machine code instructions performed during execution of the software application. The binary code is converted into machine code instructions so as to perform the inverse assembly of the compiled programming code for the software application.

20 Claims, 9 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 9

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☑ 6. Document ID: US 6279123 B1

L2: Entry 6 of 10 File: USPT Aug 21, 2001

US-PAT-NO: 6279123
DOCUMENT-IDENTIFIER: US 6279123 B1

TITLE: System for viewing and monitoring embedded processor operation

DATE-ISSUED: August 21, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Mulrooney; Timothy J.	Long Valley	NJ		

US-CL-CURRENT: 714/35; 703/20, 714/30, 714/34, 714/45, 714/726, 717/125, 717/127, 717/162

ABSTRACT:

A system for viewing and monitoring the operation of an embedded processor in which a single computer program, e.g., a Microsoft C/C++3 program, is used to test the scan chain, check the link to the target device, set up the target devices on the scan chain, load the program into the device, run the program, and display the results of the program. The present invention monitors the target processor activity, such as data flow in and out of the target processor, tracking of routines running on the target processor, and the like. Instead of relying on fixed breakpoints, the present invention utilizes breakpoints embedded in the program, enabling the user to use as many breakpoints as desired and to place them and move them wherever they are needed. The breakpoints are defined and set up in the target code (e.g. a diagnostic program) and not by the host. This allows runtime flexibility because the number and location of the breakpoints can be changed easily.

10 Claims, 6 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 6

☐ 7. Document ID: US 6090154 A

L2: Entry 7 of 10 File: USPT Jul 18, 2000

US-PAT-NO: 6090154
DOCUMENT-IDENTIFIER: US 6090154 A

TITLE: Method, apparatus and computer program product for linking stack messages to relevant information

DATE-ISSUED: July 18, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Jeffries; Robin	Palo Alto	CA		
Weatherford; David	Mountain View	CA		

Adams; Evan San Leandro CA

US-CL-CURRENT: 717/125; 717/127

ABSTRACT:

Methods, apparatus, and computer program products for linking stack messages to information relevant to stack entries and for generating the stack messages by adding link information to the stack messages. The stack messages are displayed to the user without the link information. Portions of the visible stack messages pertaining to the link information are highlighted, and upon selection by the user of the highlighted portion of the stack message, information relevant to the highlighted stack message is automatically displayed, for example, by running an editor, loading the source file corresponding to the message, and displaying the relevant portion of a source file.

40 Claims, 17 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 16

☐ 8. Document ID: US 5956479 A

L2: Entry 8 of 10 File: USPT Sep 21, 1999

US-PAT-NO: 5956479
DOCUMENT-IDENTIFIER: US 5956479 A

TITLE: Demand based generation of symbolic information

DATE-ISSUED: September 21, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
McInerney; Peter J.	Cupertino	CA		
You; Lawrence L.	San Jose	CA		
Wimble; Michael D.	Sunnyvale	CA		

US-CL-CURRENT: 714/38; 715/532, 717/127

ABSTRACT:

A human oriented object programming system (HOOPS) and its debugger provide an interactive and dynamic modeling system to assist in the incremental generation of symbolic information of computer programs which facilitates the development of complex computer programs such as operating systems and large applications with graphic user interfaces (GUIs). A program is modeled as a collection of units called components. A component represents a single compilable language element such as a class or a function. One major functionality built on HOOPS is the debugger, using symbolic properties. The database stores the components and properties. The debugger, using a GUI, displays to the user the execution state of the program. To display the execution state in terms of the programmer's source code, the debugger

demands retrieval and/or generation of the symbolic properties of the program. The compiler, which is responsible for calculating the dependencies associated with a component, uses those dependencies to generate the information stored in symbolic properties. The debugger matches versions of source and object code and retrieves source code configuration as needed. Symbolic properties that are stored in the database can be removed to reduce database and disk memory usage; they can be later reconstructed using the same method of demand-based generation of symbolic information.

24 Claims, 43 Drawing figures
Exemplary Claim Number: 9
Number of Drawing Sheets: 40

☐ 9. Document ID: US 5923881 A

L2: Entry 9 of 10 File: USPT Jul 13, 1999

US-PAT-NO: 5923881
DOCUMENT-IDENTIFIER: US 5923881 A

TITLE: Data structure display system

DATE-ISSUED: July 13, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Fujii; Kunikazu	Ayase			JP
Shinomi; Hideaki	Kawasaki			JP

US-CL-CURRENT: 717/125; 717/127, 717/131

ABSTRACT:

A program data structure is defined by a data field definition having a plurality of data field definition blocks, the data field definition blocks comprising a plurality of data items, the data items including data item names, data item hierarchy data and data item length data. A user uses a mouse pointer and clicks it on a data item in a program to select the data item. The display system of the present invention calculates offset data for selected data items. When there is a data item that is used for re-definition, the offset for a data item to be re-defined is added to the offset data to provide an offset value. The data items are arranged in order according to their hierarchical relationship, and data items that have the same offset are displayed as a single line. A numeral for an offset is added to the line, and a broken line is used to separate lines of data items.

15 Claims, 22 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 19

☐ 10. Document ID: US 5446900 A

L2: Entry 10 of 10 File: USPT Aug 29, 1995

US-PAT-NO: 5446900
DOCUMENT-IDENTIFIER: US 5446900 A

TITLE: Method and apparatus for statement level debugging of a computer program

DATE-ISSUED: August 29, 1995

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Kimelman; Paul	Walnut Creek	CA		

US-CL-CURRENT: 717/124; 714/1, 717/125, 717/127, 717/128

ABSTRACT:

A method and apparatus for statement level debugging. Statement level debugging refers to the ability to identify a statement in a source file, i.e. a breakpoint, and have program execution suspended at that point. Further, "stepping" through the execution of the program may be performed a statement at a time. In accomplishing this a column reference is added to debugging information generated in connection with an object module. The column reference provides for distinguishing the different statements in a line of a computer program. Statement level debugging allows for more precise control of the debugging of a program. Alternative means for specifying the statement are also described.

20 Claims, 8 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

Terms	Documents
L1 and (commands or opcodes)	10

Refine Search

Search Results -

Terms	Documents
L14 OR L13	13

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L15

Refine Search

Recall Text | Clear | Interrupt

Search History

DATE: Thursday, March 17, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=NO; OP=OR		
L15	L14 OR l13	13	L15
L14	L7 ANd (trojan ADJ horse) or trojanhorse	3	L14
L13	L7 ANd virus	10	L13
L12	L7 ANd (malicious ADJ code)	0	L12
L11	L10 AND ((fixed or predetermined) ADJ address)	11	L11
L10	L9 and (reference or address)	96	L10
L9	L7 and (verify or validate)	98	L9
L8	L7 and javacard	0	L8
L7	L6 and (opcode or instruction)	414	L7
L6	L4 AND microcode	543	L6
L5	(711/$$$.ccls. Or 714/$$$.ccls.)	40654	L5
L4	(711/$$$.ccls. Or 714/$$$.ccls.) AND monitor	9199	L4
L3	4266272.pn.	1	L3
L2	L1 and (commands or opcodes)	10	L2

L1	717/127.ccls. and (microcode or assembler)	22	L1

END OF SEARCH HISTORY

Refine Search

Search Results -

Terms	Documents
L10 AND ((fixed or predetermined) ADJ address)	11

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L11

Refine Search | Recall Text | Clear | Interrupt

Search History

DATE: Thursday, March 17, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=NO; OP=OR			
L11	L10 AND ((fixed or predetermined) ADJ address)	11	L11
L10	L9 and (reference or address)	96	L10
L9	L7 and (verify or validate)	98	L9
L8	L7 and javacard	0	L8
L7	L6 and (opcode or instruction)	414	L7
L6	L4 AND microcode	543	L6
L5	(711/$$$.ccls. Or 714/$$$.ccls.)	40654	L5
L4	(711/$$$.ccls. Or 714/$$$.ccls.) AND monitor	9199	L4
L3	4266272.pn.	1	L3
L2	L1 and (commands or opcodes)	10	L2
L1	717/127.ccls. and (microcode or assembler)	22	L1

END OF SEARCH HISTORY

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 13 of 13 returned.

☐ 1. Document ID: US 6795916 B2

L15: Entry 1 of 13 File: USPT Sep 21, 2004

US-PAT-NO: 6795916
DOCUMENT-IDENTIFIER: US 6795916 B2

TITLE: Method for updating microcode in a system wherein a keyboard BIOS expects an updated signal within a time period after microcode update commences

DATE-ISSUED: September 21, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Tung; Chung-Chih	Taipei			TW

US-CL-CURRENT: 713/2; 714/38, 714/55

ABSTRACT:

A novel boot method for a system, whose system BIOS has been disabled by erroneous microcode input as the result of inadvertent miscoding or malicious intent. The keyboard BIOS of the system is utilized in the present invention. If the CPU of the computer crashes during a microcode update, the system notes the error (setting a number of flags) and restarts the computer. The restarted system will not perform the microcode update, but goes straight to the normal BIOS activation, avoiding the crash problem. To fix the problem, users need only request the correct microcode from CPU retailers to amend the errors in the BIOS.

7 Claims, 2 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 2

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☐ 2. Document ID: US 6532531 B1

L15: Entry 2 of 13 File: USPT Mar 11, 2003

US-PAT-NO: 6532531
DOCUMENT-IDENTIFIER: US 6532531 B1

TITLE: Method frame storage using multiple memory circuits

DATE-ISSUED: March 11, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
O'Connor; James Michael	Mountain View	CA		
Tremblay; Marc	Palo Alto	CA		

US-CL-CURRENT: 712/202; 711/6, 712/217, 718/1

ABSTRACT:

A memory architecture in accordance with an embodiment of the present invention improves the speed of method invocation. Specifically, method frames of method calls are stored in two different memory circuits. The first memory circuit stores the execution environment of each method call, and the second memory circuit stores parameters, variables or operands of the method calls. In one embodiment the execution environment includes a return program counter, a return frame, a return constant pool, a current method vector, and a current monitor address. In some embodiments, the memory circuits are stacks; therefore, the stack management unit to cache can be used to cache either or both memory circuits. The stack management unit can include a stack cache to accelerate data transfers between a stack-based computing system and the stacks. In one embodiment, the stack management unit includes a stack cache, a dribble manager unit, and a stack control. The dribble manager unit include fill control it and a spill control unit. Since the vast majority of memory accesses to the stack occur at or near the top of the stack, the dribble manager unit maintains the top portion of the stack in the stack cache. When the stack-based computing system is popping data off of the stack and a fill condition occurs, the fill control unit transfer data from the stack to the bottom of the stack cache to maintain the top portion of the stack in the stack cache. Typically, a fill condition occurs as the stack cache becomes empty and a spill condition occurs as the stack cache becomes full.

56 Claims, 17 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 17

☐ 3. Document ID: US 6507904 B1

L15: Entry 3 of 13 File: USPT Jan 14, 2003

US-PAT-NO: 6507904
DOCUMENT-IDENTIFIER: US 6507904 B1

TITLE: Executing isolated mode instructions in a secure system running in privilege rings

DATE-ISSUED: January 14, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ellison; Carl M.	Portland	OR		

Golliver; Roger A.	Beaverton	OR
Herbert; Howard C.	Phoenix	AZ
Lin; Derrick C.	Foster City	CA
McKeen; Francis X.	Portland	OR
Neiger; Gilbert	Portland	OR
Reneris; Ken	Wilbraham	MA
Sutton; James A.	Portland	OR
Thakkar; Shreekant S.	Portland	OR
Mittal; Millind	Palo Alto	CA

US-CL-CURRENT: 712/229; 711/152, 718/100

ABSTRACT:

A technique is provided to execute isolated instructions according to an embodiment of the present invention. An execution unit executes an isolated instruction in a processor operating in a platform. The processor is configured in one of a normal execution mode and an isolated execution mode. A parameter storage containing at least one parameter to support execution of the isolated instruction when the processor is configured in the isolated execution mode.

56 Claims, 14 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 13

☐ 4. Document ID: US 6076141 A

L15: Entry 4 of 13 File: USPT Jun 13, 2000

US-PAT-NO: 6076141
DOCUMENT-IDENTIFIER: US 6076141 A
**** See image for Certificate of Correction ****

TITLE: Look-up switch accelerator and method of operating same

DATE-ISSUED: June 13, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Tremblay; Marc	Palo Alto	CA		
O'Connor; James Michael	Mountain View	CA		

US-CL-CURRENT: 711/108; 365/49, 707/6, 711/128, 711/221

ABSTRACT:

A look-up switch accelerator which includes an associative memory for storing information associated with one or more look-up switch statements. For each look-up switch statement, this information includes a look-up switch identifier value, a

plurality of match values and a corresponding plurality of jump offset values. The look-up switch accelerator also includes circuitry for determining whether a current instruction corresponds to a look-up switch statement stored in the memory, circuitry for determining whether a current match value associated with the current instruction corresponds with one of the match values stored in the memory, and circuitry for accessing a jump offset value from the memory when the current instruction corresponds to a look-up switch statement stored in the memory and the current match value corresponds with one of the match values stored in the memory (wherein the accessed jump offset value corresponds with the current match value). Also included is circuitry for retrieving match and jump offset values associated with a current look-up switch statement when the associative memory does not already contain the match and jump offset values associated with the current look-up switch statement.

17 Claims, 10 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 8

☐ 5. Document ID: US 6065108 A

L15: Entry 5 of 13 File: USPT May 16, 2000

US-PAT-NO: 6065108
DOCUMENT-IDENTIFIER: US 6065108 A
**** See image for Certificate of Correction ****

TITLE: Non-quick instruction accelerator including instruction identifier and data set storage and method of implementing same

DATE-ISSUED: May 16, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Tremblay; Marc	Palo Alto	CA		
O'Connor; James Michael	Mountain View	CA		

US-CL-CURRENT: 712/201; 711/108, 712/202

ABSTRACT:

An instruction accelerator which includes a processor and an associative memory. The processor is coupled to receive a stream of instructions and a corresponding stream of instruction identifier values. The instructions include at least one non-quick instruction which has a first associated data set which must be accessed prior to executing the non-quick instruction. A memory, which is coupled to the processor, stores one or more instruction identifier values and one or more associated data sets. The memory receives the stream of instruction identifier values. When a current instruction identifier value in the stream of instruction identifier values matches an instruction identifier value stored in the memory, an associated data set is accessed from the memory. More specifically, if the first instruction identifier value and the first data set are stored in the memory, and the current instruction identifier value is equal to the first instruction

identifier value, then the first data set is read out of the memory. Execution of the non-quick instruction is accelerated because the first data set is readily accessible within the memory. If the first data set is not stored in the memory, the associative memory and the processor control the initial retrieval of the first data set.

23 Claims, 9 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 6

☐ 6. Document ID: US 6038643 A

L15: Entry 6 of 13 File: USPT Mar 14, 2000

US-PAT-NO: 6038643
DOCUMENT-IDENTIFIER: US 6038643 A

TITLE: Stack management unit and method for a processor having a stack

DATE-ISSUED: March 14, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Tremblay; Marc	Palo Alto	CA		
O'Connor; James Michael	Mountain View	CA		

US-CL-CURRENT: 711/132

ABSTRACT:

The present invention provides a stack management unit including a stack cache to accelerate data transfers between the stack-based computing system and the stack. In one embodiment, the stack management unit includes a stack cache, a dribble manager unit, and a stack control unit. The dribble manager unit includes a fill control unit and a spill control unit. Since the vast majority of memory accesses to the stack occur at or near the top of the stack, the dribble manager unit maintains the top portion of the stack in the stack cache. Specifically, when the stack-based computing system is pushing data onto the stack and a spill condition occurs, the spill control unit transfers data from the bottom of the stack cache to the stack so that the top portion of the stack remains in the stack cache. When the stack-based computing system is popping data off of the stack and a fill condition occurs, the fill control unit transfer data from the stack to the bottom of the stack cache to maintain the top portion of the stack in the stack cache. Typically, a fill condition occurs as the stack cache becomes empty and a spill condition occurs as the stack cache becomes full.

26 Claims, 16 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 17

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☐ 7. Document ID: US 6021469 A

L15: Entry 7 of 13 File: USPT Feb 1, 2000

US-PAT-NO: 6021469
DOCUMENT-IDENTIFIER: US 6021469 A

TITLE: Hardware virtual machine instruction processor

DATE-ISSUED: February 1, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Tremblay; Marc	Palo Alto	CA		
O'Connor; James Michael	Mountain View	CA		
Joy; William N.	Aspen	CO		

US-CL-CURRENT: 711/125; 711/110, 711/118, 711/132, 712/205

ABSTRACT:

A hardware virtual machine instruction processor directly executes virtual machine instructions that are processor architecture independent. The hardware processor has high performance; is low cost; and exhibits low power consumption. The hardware processor is well suited for portable applications. These applications include, for example, an Internet chip for network appliances, a cellular telephone processor, other telecommunications integrated circuits, or other low-power, low-cost applications such as embedded processors, and portable devices.

10 Claims, 8 Drawing figures
Exemplary Claim Number: 7
Number of Drawing Sheets: 6

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☐ 8. Document ID: US 6014723 A

L15: Entry 8 of 13 File: USPT Jan 11, 2000

US-PAT-NO: 6014723
DOCUMENT-IDENTIFIER: US 6014723 A
**** See image for Certificate of Correction ****

TITLE: Processor with accelerated array access bounds checking

DATE-ISSUED: January 11, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY

Tremblay; Marc	Palo Alto	CA
O'Connor; James Michael	Mountain View	CA
Joy; William N.	Aspen	CO

US-CL-CURRENT: 711/1; 711/200

ABSTRACT:

An array boundary checking apparatus is configured to verify that a referenced element of an information array is within a maximum array size boundary value and a minimum array size boundary value. The array boundary checking apparatus of the invention includes an associative memory element that stores and retrieves a plurality of array bound values. Each one of the plurality of array bound values is associated with one of the plurality of array access instructions. An input section simultaneously compares the array access instruction identifier with at least a portion of each of the stored array reference entries, wherein the array access instruction identifier identifies an array access instruction. An output section is configured to provide as an array bounds output values one of the plurality of array bound values stored in one of the plurality of memory locations of the associated memory element. A first comparison element compares the value of the referenced element and the maximum array index boundary value and provides a maximum violation signal if the value of the element is greater than the maximum array size boundary value. A second comparison element compares the value of the element and the minimum array size boundary value and provides a minimum violation signal if the value of the element is less than the minimum array bounds value. Either a maximum violation signal or a minimum violation signal results in an exception.

11 Claims, 11 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 12

☐ 9. Document ID: US 5878256 A

L15: Entry 9 of 13 File: USPT Mar 2, 1999

US-PAT-NO: 5878256
DOCUMENT-IDENTIFIER: US 5878256 A

TITLE: Method and apparatus for providing updated firmware in a data processing system

DATE-ISSUED: March 2, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bealkowski; Richard	Delray Beach	FL		
Begun; Ralph Murray	Boca Raton	FL		
Capps, Jr.; Louis Bennie	Boynton Beach	FL		

US-CL-CURRENT: 713/2; 711/103, 717/168

ABSTRACT:

A programmable firmware store for a personal computer system includes a plurality of nonvolatile alterable electronic memories connected in a mutually paralleled circuit arrangement. The memories are connected to a controller that controls the memories to read firmware from and write firmware into the electronic memories, and to write-protect at least one of the memories. Any memory can be write-protected as initially selected by a user or technician of the system. The initial selection can be changed easily to write-protect another of the memories. The firmware in one memory includes code for checking the validity of firmware stored in another memory, and for selecting one or the other of the memories dependent upon a version code of the firmware. The controller also includes an update code for updating the firmware in a selected memory.

6 Claims, 18 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 18

☐ 10. Document ID: US 5657445 A

L15: Entry 10 of 13 File: USPT Aug 12, 1997

US-PAT-NO: 5657445
DOCUMENT-IDENTIFIER: US 5657445 A

TITLE: Apparatus and method for limiting access to mass storage devices in a computer system

DATE-ISSUED: August 12, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Pearce; John J.	DelValle	TX		

US-CL-CURRENT: 713/200; 710/37, 711/163, 712/244

ABSTRACT:

A computer system is provided with the capability of protecting portions of the mass storage media therein from unauthorized access. The mechanism employed to protect portions of the mass storage media is advantageously operating system independent. Thus, the protection mechanism functions regardless of what operating system is installed or what particular application software is presently being executed. More particularly, the computer system includes a processor configured to execute code in an operational mode and in a system management mode. A mass storage device and a memory are coupled to the processor. At least one region of the mass storage device is designated as a protected region by the user or by the manufacturer. The computer system is configured to trap mass storage device I/O operations and, in response to a trapped mass storage device I/O operation, the processor enters a system management mode. The computer system is configured to prevent execution of the trapped mass storage device I/O operation if the trapped mass storage device I/O operation is directed to a protected region of the mass

storage device. However, the computer executes the trapped mass storage device I/O operation if the trapped mass storage device I/O operation is not directed to a protected region of the mass storage device. In this manner, increased computer system security is provided to selected portions of the mass storage device without reliance on protective mechanisms within the operating system or within add-on application software.

24 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

☐ 11. Document ID: US 4525780 A

L15: Entry 11 of 13 File: USPT Jun 25, 1985

US-PAT-NO: 4525780
DOCUMENT-IDENTIFIER: US 4525780 A

TITLE: Data processing system having a memory using object-based information and a protection scheme for determining access rights to such information

DATE-ISSUED: June 25, 1985

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bratt; Richard G.	Wayland	MA		
Clancy; Gerald F.	Saratoga	CA		
Gavrin; Edward S.	Lincoln	MA		
Gruner; Ronald H.	Cary	NC		
Mundie; Craig J.	Cary	NC		
Schleimer; Stephen I.	Chapel Hill	NC		
Wallach; Steven J.	Saratoga	CA		

US-CL-CURRENT: 711/163

ABSTRACT:

A digital data processing system has a memory organized into objects containing at least operands and instructions. Each object is identified by a unique and permanent identifier code which identifies the data processing system and the object. The system uses a protection technique to prevent unauthorized access to objects by users who are identified by a subject number which identifies the user, a process of the system for executing a user's procedure, and the type of operation of the system to be performed by the user's procedure. An access control list for each object includes an access control list entry for each subject having access rights to the object and means for confirming that a particular active subject has access rights to a particular object before permitting access to the object. The system also includes stacks for containing information relating to the current state of execution of the system.

17 Claims, 200 Drawing figures

Exemplary Claim Number: 1
Number of Drawing Sheets: 156

☐ 12. Document ID: US 4498132 A

L15: Entry 12 of 13 File: USPT Feb 5, 1985

US-PAT-NO: 4498132
DOCUMENT-IDENTIFIER: US 4498132 A

TITLE: Data processing system using object-based information and a protection scheme for determining access rights to such information and using multilevel microcode techniques

DATE-ISSUED: February 5, 1985

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ahlstrom; John K.	Mountain View	CA		
Bachman; Brett L.	Boston	MA		
Belgard; Richard A.	Saratoga	CA		
Bernstein; David H.	Ashland	MA		
Bratt; Richard G.	Wayland	MA		
Clancy; Gerald F.	Saratoga	CA		
Gavrin; Edward S.	Lincoln	MA		
Gruner; Ronald H.	Cary	NC		
Jones; Thomas M.	Chapel Hill	NC		
Katz; Lawrence H.	Oregon City	OR		
Mundie; Craig J.	Cary	NC		
Richmond; Michael S.	Pittsboro	NC		
Schleimer; Stephen I.	Chapel Hill	NC		
Wallach; Steven J.	Saratoga	CA		
Wallach, Jr.; Walter A.	Raleigh	NC		
Wells; Douglas M.	Chapel Hill	NC		

US-CL-CURRENT: 711/163

ABSTRACT:

A digital data processing system has a memory organized into objects containing at least operands and instructions. Each object is identified by a unique and permanent identifier code which identifies the data processing system and the object. The system further uses multilevel microcode techniques for controlling sequences of microinstructions and for controlling the interval operations of the processor. The system uses a protection technique to prevent unauthorized access to objects by users who are identified by a subject number which identifies the user, a process of the system for executing a user's procedure, and the type of operation of the system to be performed by the user's procedure. An access control list for each object includes an access control list entry for each subject having access

rights to the object and means for confirming that a particular active subject has access rights to a particular object before permitting access to the object. The system also includes stacks for containing information relating to the current state of execution of the system.

47 Claims, 1 Drawing figures
Exemplary Claim Number: 34
Number of Drawing Sheets: 1

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☐ 13. Document ID: US 4328542 A

L15: Entry 13 of 13 File: USPT May 4, 1982

US-PAT-NO: 4328542
DOCUMENT-IDENTIFIER: US 4328542 A

TITLE: Secure implementation of transition machine computer

DATE-ISSUED: May 4, 1982

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Anastas; Mark S.	Auburn	WA		
Vaughan; Russell F.	Enumclaw	WA		

US-CL-CURRENT: 711/163; 711/148

ABSTRACT:

A secure implementation of a transition machine utilizing requirements oriented application programming and a hardware executive. The hardware executive is physically separate and protected from data processors executing the application programs and limits access authorization both for program execute and data read and write operations.

46 Claims, 20 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 13

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
L14 OR L13	13